**Report - Fine art galleries**

**Student: Jan Dzięcielski**

**Course: Data Engineering group II**

**Subject Analysis**

Purpose of creating this database is to store information about art transactions during auctions held by galleries.

Customer is the owner of galleries who is concerned about proper data storage inside the business. User of the database is a person working for a particular gallery that is responsible for keeping track of the auctions, participants, and artworks that are stored in the galleries.

Possible scenarios of using this database are:

* keeping track of bids for particular artwork and artists;
* storing up-to-date information about stored and auctioned artwork;
* who sold and who bought the artwork;
* how many people participated in the auction.

This database is not supposed to keep track of day-to-day functioning of galleries. The major point of creating this database is to store information about transactions made in particular galleries by participants of the auctions. Goal of having this database is to track the efficiency and quality of service provided by fine art galleries’ auctions.

**ERD Diagram**



**Description of ERD diagram**

**Entities and their attributes:**

* **Galleries** - set of data related to a single gallery, changes related to stored artwork are applied only after finished auctions. New entry added only when a new gallery opens. ~40 entries.
* address - name of the street, number of street that gallery is placed in (text, english alphabet)
* city - name of the city that a gallery is placed in (text, english alphabet)
* name of the gallery - unique official name of the gallery (text, english alphabet)
* number of artwork - number of currently stored artworks (int)
* number of previous auctions - number of previously held auctions (int)
* sold artworks - number of sold artworks by a gallery (int)
* **Stored** - consists of information of artwork stored by a particular gallery. ~1500 entries
* starting date of storage - the date that the artwork began to be stored in a gallery. (date, DD-MM-YYYY)
* ending date of storage - the date that the artwork ended its storage in a gallery. (date, DD-MM-YYYY)
* name of the gallery (FK) - name of the gallery that an artwork was being stored in, foreign key of the Galleries entity.
* city (FK) - city that the gallery is placed in, foreign key of the Galleries entity.
* name of the artwork (FK) - name of an artwork that was being stored, foreign key of the Artworks entity.
* date of creation (FK) - date of creation of the artwork, foreign key of the Artworks entity.
* **Artworks** - description about stored artwork in the gallery. Artwork may belong to the gallery but it may as well belong to an individual that requested storage. ~1000 entries
* name of the artwork - name of an artwork (text, english alphabet)
* date of creation - date that an artwork was made (date, DD:MM:YYYY)
* style - a style that the artwork was made in (text, english alphabet)
* period in art - style in which the artwork was made (text, english alphabet)
* ID - unique combination of a name of an artwork, the gallery that an artwork is stored in, and the owner (text, english alphabet)
* name of the author (FK)
* date of birth (FK)
* **Transactions** (finalized) - completed transactions from auctions. Contains information about who bought and who sold the artwork. ~800 entries
* date - date that the transaction was made (date, DD-MM-YYYY)
* time - time of finalized transaction (time, HH:MM:SS)
* ID - unique combination of a name of a buyer, name of a seller, and date (text, english alphabet)
* name of participant - name of a buyer/seller (text, english alphabet)
* name of the gallery (FK) - name of the gallery that a transaction is being finalized (text, english alphabet)
* city (FK)
* **Authors** - information about the authors of created artwork. ~700 entries
* ID - unique combination of name, surname and date of birth (text, english alphabet)
* date of birth - date of birth of an author (date, DD-MM-YYYY)
* name - name of an author (text, english alphabet)
* surname - surname of an author (text, english alphabet)
* **Contracts** - contracts with individual private clients that applied for storing their artwork(s). ~500 entries
* starting date of storing - day that the artwork began its storage (date, DD-MM-YYYY)
* ending date of storing - day that the artwork endedits storage (date DD-MM-YYYY)
* name of the gallery (FK) - name of the gallery that the artwork was being stored in, foreign key from entity Galleries (text, english alphabet)
* name of the artwork (FK) - name of the particular artwork, foreign key from Artworks entity (text, english alphabet)
* name of the owner (FK) - name and surname of the owner of the artwork, foreign key from Owners entity (text, english alphabet)
* telephone number (FK) - telephone number to a client, foreign key from Owners entity (7-9 digits, int)
* **Owners** - entity consisting of information about an individual client. ~120 entries
* name of the owner - name and surname of the owner (text, english alphabet)
* telephone number - telephone number to an owner (7-9 digits, int)
* email - contact email to an owner (text, english alphabet)
* **Starting price** - starting price of an auction for a particular artwork. ~6000 entries
* name of the artwork (FK) - name of an artwork, foreign key from Artworks entity (text, english alphabet)
* date of creation (FK) - date of creation of a particular artwork, foreign key of Artworks entity (date, DD-MM-YYYY)
* starting price - starting price of an auction, (int, price in $)
* last price - price that ended the auction, the artwork was sold at this price (int, price in $)
* date (FK) - date that an auction was held in, foreign key from Auctions entity (date DD-MM-YYYY)
* ID Auction (FK) - ID of an auction, foreign key of Auctions entity (int)
* hour of the bid (FK) - hour at which the betting began (time, HH:MM:SS)
* **Auctions** - set of information concerning the auction held on a particular day. Added as soon as possible after the auction has finished. ~1200 entries
* date - date that the particular auction was held (date, DD-MM-YYYY)
* ID - unique combination of number of an auction and name of the gallery (text, english alphabet)
* started at (hour) - hour of the start of an auction (time, HH:MM:SS)
* finished at (hour) - hour of the end of an auction (time, HH:MM:SS)
* sales made - sales made during an auction (int)
* guests invited - number of people that were invited to an auction (int)
* guests appeared - number of people that showed up for an auction (int)
* number of finalized sales - number of finalized sales during an auction (int)
* name of the gallery (FK) - name of the gallery that auction was held in, foreign key from Galleries entity (text, english alphabet)
* **Offers** - offers made by participants of auction (bidders). Most important information is the price set by a bidder, and the time of a bid. It will be used to determine how much the participants were interested in a particular artwork, style, or an artist. ~3000 entries
* price - amount of money that a bidder offered for an artwork (int)
* hour of bid - exact hour of a bid (time, HH:MM:SS)
* ID - unique combination of number of an auction and name of the gallery (text, english alphabet)
* name of the participant (FK) - name of a participant that bidded, foreign key from Participants entity (text, english alphabet)
* country (FK) - country that the bidder is from, foreign key from Participants entity.
* **Participants** - information about participants of the auctions. ~700 entries
* email - contact email to a participant (text, english alphabet, @ symbol)
* phone - contact phone to a participant (int, 7-9 digits)
* ID - composition of name, surname, country and phone number (text, english alphabet)
* name of the participant - name and surname of the participant (text, english alphabet)
* country - country that a participant comes from (text, english alphabet)

**Description of relationships:**

* conducted (entities: Galleries & Auctions)
* relation: 1 : 0..n
* galleries that conducted auctions, each gallery may have conducted none or many auctions, but each auction had to be conducted by a one gallery
* offered by (entities: Offers & Participants)
* relation: 0..n : 1
* Participants giving offers during auctions, many or none offers can be made, an offer always needs to have on participant.
* created by (entities: Authors & Artworks)
* relation: 1..n : 1..n
* Artworks created by Authors, each artwork needs to have one or more authors, each authors have one or more artworks registered.
* sold (entities: Transactions (finalized) & Participants)
* relation: 1..n : 1
* A participant may have many transactions, each transaction always was made by one participant
* bought (entities: Transactions (finalized) & Participants)
* relation: 1..n : 1
* A participant may have many finalized transactions, each transaction always was made by one participant
* were (entities: Artworks & Stored)
* relation: 1 : 0..n
* Artworks that were being stored, many artworks may be stored, but each stored status is signed to particular artwork
* in (entities: Stored & Galleries)
* relation: 0..n : 1
* Each Stored status will belong to one particular gallery, while galleries will have many or none stored statuses
* began at (entities: Starting price & Artworks)
* relation: 1 : 1
* Each artwork will have one starting price and each starting price is assigned to one artwork
* started (entities: Auctions & Starting price)
* relation: 1..n : 1
* Each auction will have many starting prices due to many artworks being auctioned, one starting price will always relate to one auction.
* made (entities: Starting price & Offers)
* relation: 0..n : 1
* offers can be made multiple time or not at all after starting price, a starting price always relates to a particular offer
* on (entities: Artworks & Contracts (for storing))
* relation: 0..n : 0..1
* relates to artworks being stored on contracts for clients, each artwork will relate to one contract, but many artworks can be signed on one contract
* had (entities: Transactions (finalized) & Galleries)
* relation: 0..n : 1
* each transaction will be related to a particular gallery that it was made in, but galleries will have multiple transactions made
* had (entities: Contracts (for storing) & Owners)
* relation: 0..n : 1
* describes owners having multiple or none contracts for storing, contract always will have one owner
* finished (entities: Starting price & Transactions (finalized))
* relation: 1 : 1
* each starting price will result in one particular finalized transaction and each transaction will relate to one starting price
* held in (entities Galleries & Contracts (for storing) )
* relation: 1 : 0..n
* many or none contracts will relate to a particular gallery due to storage in that gallery.

**Relational Database Schema (RDS):**

* Galleries(address, city, name of the gallery, number of artwork, sold artworks)
* Auctions(date, ID, started at (hour), ended at(hour), sales made, guests invited, guests appeared, number of finalized sales, name of the gallery (FK), city (FK))
* Galleries(address, city, name of the gallery, number of artwork, sold artworks)
* Contracts(starting date of storing, ending date of storing, name of the gallery(FK), city(FK), name of the artwork(FK), name of the owner(FK), telephone number(FK))
* Galleries(address, city, name of the gallery, number of artwork, sold artworks)
* Transactions(date, time, ID, name of the participant(FK), name of the gallery (FK), city (FK))
* Galleries(address, city, name of the gallery, number of artwork, sold artworks)
* Stored(starting date of storage, ending date of storage, name of the gallery(FK), city(FK), name of the artwork(FK), date of creation(FK))
* Auctions(date, ID, started at (hour), ended at(hour), sales made, guests invited, guests appeared, number of finalized sales, name of the gallery (FK), city (FK)
* Starting price(name of the artwork (FK), date of creation(FK), starting price, last price, date(FK), ID Auction(FK), hour of the bid(FK))
* Artworks(name of the artwork, date of creation, style, period in art, ID, name of the author (FK), date of birth(FK))
* Stored(starting date of storage, ending date of storage, name of the gallery(FK), city(FK), name of the artwork(FK), date of creation(FK))
* Artworks(name of the artwork, date of creation, style, period in art, ID, name of the author (FK), date of birth(FK))
* Contracts(starting date of storing, ending date of storing, name of the gallery(FK), name of the artwork(FK), name of the owner(FK), telephone number(FK))
* Artworks(name of the artwork, date of creation, style, period in art, ID, name of the author (FK), date of birth(FK))
* Starting price(name of the artwork (FK), date of creation(FK), starting price, last price, date(FK), ID Auction(FK), hour of the bid(FK), ID Artworks)
* Artworks(name of the artwork, date of creation, style, period in art, ID, name of the author (FK), date of birth(FK))
* Authors(ID, date of birth, date of death, name of the author)
* Artworks\_Authors(ID REF Authors,ID REF Artworks)
* Contracts(starting date of storing, ending date of storing, name of the gallery(FK), name of the artwork(FK), name of the owner(FK), telephone number(FK))
* Owners(name of the owner, telephone number, email)
* Starting price(name of the artwork (FK), date of creation(FK), starting price, last price, date(FK), ID Auction(FK), hour of the bid(FK), ID REF Offers)
* Offers(price, hour of the bid, ID, name of the participant(FK), country (FK))
* Offers(price, hour of the bid, ID, name of the participant(FK), country (FK))
* Participants(email, phone, ID, name of the participant, country)
* Participants(email, phone, ID, name of the participant, country)
* Transactions(date, time, ID, name of the participant(FK), country(FK) name of the gallery (FK), city (FK))